Performance Characterization and Transmission Schemes for Instantly Decodable Network Coding in Wireless Broadcast

Mingchao Yu, Parastoo Sadeghi, and Neda Aboutorab

Research School of Engineering, The Australian National University, Canberra, Australia

Email: {ming.yu,parastoo.sadeghi,neda.aboutorab}@anu.edu.au

Abstract

We consider broadcasting a block of packets to multiple wireless receivers under random packet erasures using instantly decodable network coding (IDNC). The sender first broadcasts each packet uncoded once, then generates coded packets according to receivers' feedback about their missing packets. We focus on strict IDNC (S-IDNC), where each coded packet includes at most one missing packet of every receiver. But we will also compare it with general IDNC (G-IDNC), where this condition is relaxed. We characterize two fundamental performance limits of S-IDNC: 1) the number of transmissions to complete the broadcast, and 2) the average delay for a receiver to decode a packet. We derive a closed-form expression for the expected minimum number of transmissions in terms of the number of packets and receivers and the erasure probability. We prove that it is NP-hard to minimize the decoding delay of S-IDNC. We also derive achievable upper bounds on the above two performance limits. We show that G-IDNC can outperform S-IDNC without packet erasures, but not necessarily with packet erasures. Next, we design optimal and heuristic S-IDNC transmission schemes and coding algorithms with full/intermittent receiver feedback. We present simulation results to corroborate the developed theory and compare with existing schemes.

Index terms– Wireless broadcast, network coding, throughput, decoding delay, instantly decode.

I. Introduction

The broadcast nature of wireless medium allows one sender to simultaneously serve multiple receivers who are interested in the same data. We consider a block-based wireless broadcast system where a sender wishes to deliver a block of data packets to a set of receivers, with the channels between the sender and the receivers are subject to independent random packet erasures. A traditional approach is to send the data packets unaltered under a receiver feedback mechanism, such as Automatic-Repeat-reQuest (ARQ)

[1]. This approach, though simple, is inefficient in terms of throughput, as the transmitted packets are non-innovative to the receivers who have already received them.

The advent of network coding [2] starts a new era for high throughput network coded wireless communications [3]–[17]. By linearly adding all data packets together with randomly chosen coefficients from a sufficiently large finite field, random linear network coding (RLNC) can almost surely achieve the optimal block completion time in block-based wireless broadcast [9]–[11], [18], which is defined as the number of transmissions it takes to complete the broadcast, and is a fundamental measure of throughput due to their inverse relation under a fixed block size. Compared to other optimal codes such as Fountain codes [19], [20], RLNC is preferred due to its ease of implementation at the sender and extension to more complex networks and traffic settings.

However, with RLNC, data packets are block-decoded by solving a set of linear equations, which only takes place after a sufficient number of coded packets have been received. RLNC thus may suffer from heavy computational load [11] and packet decoding delay [17], which is measured by the average time it takes for a receiver to decode a data packet. The first issue can, for example, hinder the application of RLNC for mobile receivers with limited computational capability [11]. Meanwhile, a large packet decoding delay can be unacceptable for delay-sensitive applications such as video streaming [21], [22].

To mitigate these issues, instantly decodable network coding (IDNC) techniques [3], [12]–[15] have been introduced. With IDNC, the sender first broadcasts the data packets uncoded once. It then makes online coding decisions based on receivers' feedback about their packet reception state, under the restriction that coding/decoding is over binary field. A simple packet reception state is demonstrated in Table I. There are two data packets, $\mathbf{p}_1$ and $\mathbf{p}_2$, and three receivers, R_1 to R_3, where each has a subset of $\{\mathbf{p}_1, \mathbf{p}_2\}$ and wants the rest. Consider a coded packet of $\boldsymbol{X} = \mathbf{p}_1 \oplus \mathbf{p}_2$, where $\oplus$ denotes binary XOR. It has three different effects on different receivers: 1) it is instantly decodable to R_1, because R_1 can decode $\mathbf{p}_2$ by performing $\boldsymbol{X} \oplus \mathbf{p}_1$; 2) it is non-instantly decodable to R_2, because R_2 has neither $\mathbf{p}_1$ nor $\mathbf{p}_2$; and 3) it is non-innovative to R_3, because R_3 already has both $\mathbf{p}_1$ and $\mathbf{p}_2$.

There are two main types of IDNC techniques. The first one, called strict IDNC (S-IDNC) [13]–[15], prohibits the transmissions of non-instantly decodable packets to any receiver. Effectively, each coded

TABLE I

A SIMPLE PACKET RECEPTION STATE FOR IDNC CODING

	$\mathbf{p}_1$	$\mathbf{p}_2$
R_1	has	wants
R_2	wants	wants
R_3	has	has

packet can include at most one wanted data packet of every receiver. The second one, called general IDNC (G-IDNC), removes this restriction to generate more coding opportunities.

There is a large body of research on G-IDNC, focusing on its throughput and decoding delay performance, coding algorithms, and transmission schemes. Early models and heuristics related to G-IDNC were proposed for index coding [23]. Then G-IDNC was introduced for wireless broadcast and was graphically modeled in [12]. Although its best performance remains unidentified, powerful heuristic algorithms have been developed to improve its throughput and/or decoding delay [12], [24]–[26], or to strike a balance between the two [27]. G-IDNC transmission schemes under full/intermittent receiver feedback have been developed [28], [29]. G-IDNC has also been adopted in wireless broadcast applications with hard decoding deadlines [21] or with receiver cooperation [30]–[32].

Studies on theoretical performance characterization and implementations of S-IDNC are more limited in both range and depth. S-IDNC was graphically modeled in [13], which then proved that the minimum clique partition solution [13] of the associated graph can be an S-IDNC solution that minimizes the block completion time. However, this solution does not take into account the issues of decoding delay and the robustness of coded transmissions to erasures. S-IDNC has shown to be asymptotically throughput optimal when there are up to three receivers or when the number of data packets approaches infinity [21], but the general relation between the throughput of S-IDNC and system parameters has not been characterized before. In addition and to the best of our knowledge, the minimum packet decoding delay of S-IDNC is still unknown. Moreover, there have not been S-IDNC transmission schemes that can work with intermittent receiver feedback. Another unaddressed problem is a systematic performance comparison between S-IDNC and G-IDNC.

In this paper, we study the above problems and provide the following contributions:

1) We characterize the throughput performance limits of S-IDNC. Specifically, we first derive an achievable upper bound on the minimum block completion time for any given packet reception state. We then derive a closed-form expression for the expected minimum block completion time in terms of the number of packets and receivers and the erasure probability;
2) We prove that it is NP-hard to minimize the packet decoding delay of S-IDNC. We derive an upper bound on the minimum packet decoding delay in terms of the minimum block completion time;
3) We show that in the presence of erasures, the minimum clique partition solution of the S-IDNC graph as identified by [13] may not result in the minimum block completion time, because it does not allow the same data packet to be repeated in different coded packets, a desired property which we refer to as packet diversity. Motivated by this fact, we develop the optimal S-IDNC coding algorithm, as well as heuristics that aim to improve packet diversity. We design S-IDNC transmission schemes under full and intermittent receiver feedback;
4) We also provide new results on how S-IDNC and G-IDNC compare. We study the relation between S-IDNC and G-IDNC graphs, and then demonstrate some scenarios under which G-IDNC can/cannot outperform S-IDNC.

II. System Model and Notations

A. Transmission Setup

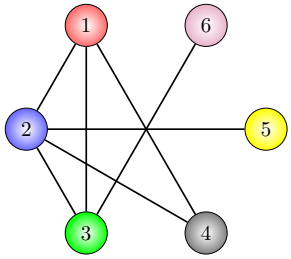
We consider a block-based wireless broadcast scenario, in which the sender needs to deliver a block of K data packets, denoted by $\mathcal{P}_K = \{\mathbf{p}_k\}_{k=1}^{K}$, to N receivers, denoted by $\mathcal{R}_N = \{R_n\}_{n=1}^{N}$ through wireless channels that are subject to independent random packet erasures.

Initially, the K data packets are transmitted uncoded once using K time slots, constituting a *systematic transmission phase* [11]. Then, each receiver provides feedback[1] to the sender about the packets it has received or missed due to packet erasures. The complete packet reception state is represented by an $N \times K$ state feedback matrix (SFM) $\boldsymbol{A}$, where $a_{n,k} = 1$ if R_n has missed (and thus still wants) $\mathbf{p}_k$, and

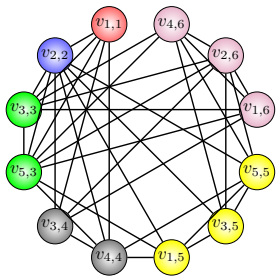
[1]We assume that there exists an error-free feedback link from each receiver to the sender that can be used with appropriate frequency.

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$
R_1	1	0	0	0	1	1
R_2	0	1	0	0	0	1
R_3	0	0	1	1	1	0
R_4	0	0	0	1	0	1
R_5	0	0	1	0	1	0

(a) State feedback matrix $\boldsymbol{A}$



(b) S-Graph $\mathcal{G}_s$



(c) G-Graph $\mathcal{G}_g$

Fig. 1. An example of SFM and its S- and G-IDNC graphs.

$a_{n,k} = 0$ if R_n has already received $\mathbf{p}_k$. The set of data packets wanted by R_n is called the *Wants* set of R_n and is denoted by $\mathcal{W}_n$. The set of receivers who want $\mathbf{p}_k$ is called the *Target* set of $\mathbf{p}_k$ and is denoted by $\mathcal{T}_k$. The size of $\mathcal{T}_k$ is denoted by T_k. Packets with larger T_k are more desired by receivers.

Example 1. *Consider the SFM in Fig. 1(a) with $K = 6$ data packets and $N = 5$ receivers. The Wants set of R_1 is $\mathcal{W}_1 = \{\mathbf{p}_1, \mathbf{p}_5, \mathbf{p}_6\}$. The Target set of $\mathbf{p}_3$ is $\mathcal{T}_3 = \{R_3, R_5\}$ and thus $T_3 = 2$.*

B. Coded Transmission Phase: Two Types of IDNC

According to $\boldsymbol{A}$, the sender generates IDNC coded packets under the binary field $\mathbb{F}_2$. Explicitly, IDNC coded packets are of the form $\boldsymbol{X} = \bigoplus_{\mathbf{p}_k \in \mathcal{M}} \mathbf{p}_k$, where $\mathcal{M}$ is a selected subset of $\mathcal{P}_K$, and is called an IDNC coding set. Obviously, $\boldsymbol{X}$ has three possible decoding effects at each receiver:

Definition 1.1. *An IDNC coded packet $\boldsymbol{X}$ is instantly decodable for receiver R_n if $\mathcal{M}$ contains only one data packet from the Wants set $\mathcal{W}_n$ of R_n, i.e., if $|\mathcal{M} \cap \mathcal{W}_n| = 1$.*

Definition 1.2. *An IDNC coded packet $\boldsymbol{X}$ is non-instantly decodable for receiver R_n if $\mathcal{M}$ contains two or more data packets from the Wants set $\mathcal{W}_n$ of R_n, , i.e., if $|\mathcal{M} \cap \mathcal{W}_n| > 1$.*

Definition 1.3. *An IDNC coded packet $\boldsymbol{X}$ is non-innovative for receiver R_n if $\mathcal{M}$ contains no data packets from the Wants set $\mathcal{W}_n$ of R_n, , i.e., if $|\mathcal{M} \cap \mathcal{W}_n| = 0$. Otherwise, it is innovative.*

Depending on which of the above three types of effects are allowed, there are two variations of IDNC. The first one is called strict IDNC (S-IDNC), which is the main subject of our study. It prohibits the transmission of any non-instantly decodable coded packets to any receivers. This restriction implies that

any two data packets wanted by the same receiver cannot be coded together. We thus have the concept of conflicting and non-conflicting data packets:

Definition 2. *Two data packets* $\mathbf{p}_i$ *and* $\mathbf{p}_j$ *conflict if at least one receiver wants both of them, i.e., if* $\exists n : \{\mathbf{p}_i, \mathbf{p}_j\} \subseteq \mathcal{W}_n$. *Otherwise they do no conflict.*

An S-IDNC coding set is thus a set of pairwise non-conflicting data packets. The conflicting states among data packets can be represented by an undirected graph $\mathcal{G}_s(\mathcal{V}, \mathcal{E})$. Each vertex $\boldsymbol{v}_i \in \mathcal{V}$ represents a data packet $\mathbf{p}_i$. Two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are connected by an edge $\boldsymbol{e}_{i,j} \in \mathcal{E}$ if $\mathbf{p}_i$ and $\mathbf{p}_j$ do not conflict. Thus, every complete subgraph of $\mathcal{G}_s$, a.k.a., a clique, represents an S-IDNC coding set. In the rest of the paper, we will use the terms "coded packet", "coding set", and "clique" interchangeably, and denote the last two by $\mathcal{M}$.

The main limitation of S-IDNC is that a coded packet which is instantly decodable for a large subset of receivers may be prohibited merely because it is non-instantly decodable for a small subset of receivers. In the second type of IDNC, called general IDNC (G-IDNC), the restriction on non-instantly decodable packets is removed to generate more coding opportunities.

G-IDNC can also be graphically modeled [24]. The difference is that, in the G-IDNC graph $\mathcal{G}_g(\mathcal{V}, \mathcal{E})$, a data packet $\mathbf{p}_k$ wanted by different receivers are individually represented by different vertices $\boldsymbol{v}_{n,k}$, for all $a_{n,k} = 1$. Consequently, the number of vertices in $\mathcal{G}_g$ is equal to the number of "1"s in $\boldsymbol{A}$. Two vertices $\boldsymbol{v}_{m,i}$ and $\boldsymbol{v}_{n,j}$ are connected by an edge if: 1) $i = j$, or 2) if $\mathbf{p}_i \notin \mathcal{W}_n$ and $\mathbf{p}_j \notin \mathcal{W}_m$. In the first case, $\mathbf{p}_i = \mathbf{p}_j$, and thus by sending $\mathbf{p}_i$ both R_m and R_n can decode. In the second case, by sending $\mathbf{p}_i \oplus \mathbf{p}_j$, R_m and R_n can decode $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively, because they already have $\mathbf{p}_j$ and $\mathbf{p}_i$, respectively. Similar to S-IDNC, every clique of $\mathcal{G}_g$ represents a G-IDNC coding set.

We note that an S-IDNC coded packet is always a G-IDNC coded packet, but the reverse is not necessarily true. Below is an example of S- and G-IDNC coded packets.

Example 2. *Consider the SFM and its S- and G-IDNC graphs in Fig. 1. The G-IDNC graph indicates that* $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{5,3}, \boldsymbol{v}_{4,4})$ *is a clique. The corresponding G-IDNC coding set is* $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4)$*, and thus* $\boldsymbol{X}_g = \mathbf{p}_1 \oplus \mathbf{p}_3 \oplus \mathbf{p}_4$ *is a G-IDNC coded packet.* $\boldsymbol{X}_g$ *is instantly decodable for* R_1, R_4, R_5 *because they only*

want one data packet from $\boldsymbol{X}_g$. $\boldsymbol{X}_g$ is non-instantly decodable for R_3 because R_3 wants both $\mathbf{p}_3$ and $\mathbf{p}_4$. $\boldsymbol{X}_g$ is non-innovative for R_2.

Due to the existence of R_3, $\boldsymbol{X}_g$ is not an S-IDNC coded packet. Whereas the S-IDNC graph indicates that $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ is a clique. The corresponding coding set is $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$, and thus $\boldsymbol{X}_s = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \mathbf{p}_3$ is an S-IDNC coded packet, which can be verified to also correspond to clique $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{3,3})$ or clique $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{5,3})$ in the G-IDNC graph.

We then introduce the notion of *IDNC solution*. A set of IDNC coding sets is called an IDNC solution if, upon the reception of the coded packets of all these coding sets, every receiver can decode all its wanted data packets. An S-IDNC solution is denoted by $\mathcal{S}_s$. The set of all S-IDNC solutions of a given SFM is denoted by $\mathbb{S}_s$. Similarly, we can also define $\mathcal{S}_g$ and $\mathbb{S}_g$ for G-IDNC.

For the SFM in Fig. 1, by partitioning the S-IDNC graph into three disjoint cliques, we can obtain, among others, an S-IDNC solution of $\mathcal{S}_s = \{(\mathbf{p}_1, \mathbf{p}_4), (\mathbf{p}_2, \mathbf{p}_5), (\mathbf{p}_3, \mathbf{p}_6)\}$. Similarly, a disjoint clique partition of the G-IDNC graph is $\{(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{5,3}, \boldsymbol{v}_{4,4})$, $(\boldsymbol{v}_{3,3}, \boldsymbol{v}_{1,6}, \boldsymbol{v}_{2,6}, \boldsymbol{v}_{4,6})$, $(\boldsymbol{v}_{1,5}, \boldsymbol{v}_{3,5}, \boldsymbol{v}_{5,5})$, $(\boldsymbol{v}_{3,4}, \boldsymbol{v}_{4,4})\}$, indicating a G-IDNC solution of $\mathcal{S}_g = \{(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4), (\mathbf{p}_3, \mathbf{p}_6), (\mathbf{p}_5), (\mathbf{p}_4)\}$.

To assess the performance of IDNC solutions, we now introduce our measures of throughput and decoding delay.

C. Throughput and Decoding Delay Measures

An S-IDNC solution $\mathcal{S}_s$ requires a minimum of $|\mathcal{S}_s|$ coded transmissions. We call $U_{\mathcal{S}_s} \triangleq |\mathcal{S}_s|$ the *minimum block completion time* of $\mathcal{S}_s$. It measures the best throughput of $\mathcal{S}_s$ with a value of $\frac{K}{K+U_{\mathcal{S}_s}}$ packet per transmission. We further denote by U_s the absolute minimum block completion time over all the S-IDNC solutions of $\boldsymbol{A}$, i.e., $U_s \triangleq \min\{U_{\mathcal{S}_s} : \mathcal{S}_s \in \mathbb{S}_s\}$. The definition of U_g for G-IDNC follows.

We measure decoding delay by *average packet decoding delay* (APDD) D, which is the average time it takes for a receiver to decode a data packet:

$$D = \frac{1}{T} \sum_{\forall a_{n,k}=1} u_{n,k} \tag{1}$$

where $u_{n,k}$ is the time index when R_n decodes $\mathbf{p}_k$, and $T = \sum_{k=1}^{K} T_k$, which is also the number of "1"s in $\boldsymbol{A}$. Given an IDNC solution $\mathcal{S}$, by letting $u_{n,k}$ be the *first* time index when $\mathcal{S}$ allows R_n to

TABLE II

THE DECODING DELAY OF ORIGINAL DATA PACKETS AT THE RECEIVERS

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$
R_1	1	0	0	0	4	2
R_2	0	1	0	0	0	2
R_3	0	0	2	3	4	0
R_4	0	0	0	3	0	2
R_5	0	0	2	0	4	0

decode $\mathbf{p}_k$, (1) produces the minimum APDD of $\mathcal{S}$. Then similar to U_s (resp. U_g), we denote by D_s (resp. D_g) the absolute minimum APDD over all S- (resp. G-) IDNC solutions of $\boldsymbol{A}$. We also note that in the specific case of an S-IDNC solution $\mathcal{S}_s$, $u_{n,k}$ is indeed the index of the first coding set in $\mathcal{S}_s$ that contains $\mathbf{p}_k$, as every receiver who wants $\mathbf{p}_k$ can decode it from this coding set.

Example 3. *Consider the SFM in Fig. 1(a). Suppose that an S-IDNC solution with four coded packets* $\boldsymbol{X}_1 = \mathbf{p}_1 \oplus \mathbf{p}_2$, $\boldsymbol{X}_2 = \mathbf{p}_3 \oplus \mathbf{p}_6$, $\boldsymbol{X}_3 = \mathbf{p}_4$, *and* $\boldsymbol{X}_4 = \mathbf{p}_5$ *are transmitted in this order. The receivers' decoding time* $\{u_{n,k}\}$ *are summarized in Table II. The minimum APDD of this solution is* $D_{\mathcal{S}_s} = (1 \times 2 + 2 \times 5 + 3 \times 2 + 4 \times 3)/12 = 2.5$.

In the presence of packet erasures, the sender needs to adopt a *coded transmission phase*. In each coded transmission, it selects and broadcasts a coding set through erasure-prone wireless channels. We denote by U_T the block completion time of this phase, and by D_T the APDD of this phase, calculated as in (1). U_T and D_T measure the throughput and decoding delay performance of this phase, respectively. They vary according to the IDNC solutions, transmission schemes, and erasure patterns. But it always holds that $U_T \geqslant U_s$ and $D_T \geqslant D_s$ if S-IDNC is applied. Therefore, U_s and D_s reflect the performance limits of S-IDNC. Hence, we will first study these limits in the next section, and then design S-IDNC transmission schemes and coding algorithms in Sections IV and V, respectively.

III. PERFORMANCE LIMITS AND PROPERTIES OF IDNC

In this section, we study performance limits and properties of S-IDNC and compare it with G-IDNC.

A. Absolute minimum block completion time U_s

We first study the throughput limit of S-IDNC, measured by the absolute minimum block completion time U_s. It has been proved that U_s is equal to the size of the minimum clique partition solution[2] of $\mathcal{G}_s$ [13], denoted by $\mathcal{S}_c$. This equivalence holds because of the following property:

Property 1. *Removing any vertex from the S-IDNC graph does not change the connectivity of the remaining vertices.*

This property holds because vertices in $\mathcal{G}_s$ represent different data packets. Thus, to remove all vertices from $\mathcal{G}_s$ (i.e., to complete the broadcast), at least $|\mathcal{S}_c|$ cliques must be removed, which yields $U_s = |\mathcal{S}_c|$.

According to graph theory, $|\mathcal{S}_c|$ is equal to the chromatic number[3] $\chi(\overline{\mathcal{G}}_s)$ of the complementary graph $\overline{\mathcal{G}}_s$, which has the same vertex set as $\mathcal{G}_s$, but has opposite vertex connectivity. We thus have $U_s = \chi(\overline{\mathcal{G}}_s)$. This equality enables us to answer two important questions about U_s: 1) how to find the U_s of a given S-IDNC graph? 2) what are the statistical characteristics of U_s under random packet erasures?

1) The U_s of an SFM: The chromatic number of a graph (and thus U_s) has been proven to be NP-hard to find and AXP-hard to approximate [33]. But there are heuristic algorithms and bounds for it. We will develop algorithms dedicated to S-IDNC in Section V, and focus on the bounds in this subsection.

Tight bounds on $\chi(\overline{\mathcal{G}}_s)$ exist, but are also NP-hard to find. One such example is a tight lower bound $w(\overline{\mathcal{G}}_s)$ [34], the size of the largest clique of $\overline{\mathcal{G}}_s$. There are also loose bounds. For example:

Property 2. *All S-IDNC graphs with K vertices and M_0 edges have:*

$$U_s \geqslant \left\lceil K^2/(K+2M_0) \right\rceil \tag{2}$$

where $\lceil x \rceil$ outputs the smallest integer greater than x.

This bound is due to Geller [35] and its proof is omitted here. This bound is useful because it identifies the smallest *achievable* U_s of all the S-IDNC graphs with K vertices and M_0 edges.

[2]The minimum clique partition solution of a graph $\mathcal{G}$ is the minimum set of disjoint cliques of $\mathcal{G}$ that together cover all the vertices.

[3]The chromatic number of a graph $\mathcal{G}$ is the minimum number of colors to color the vertices so that any two connected vertices have different colors.

An existing upper bound on $\chi(\overline{\mathcal{G}}_s)$ is $\Delta(\overline{\mathcal{G}}_s)+1$ [33], where $\Delta(\overline{\mathcal{G}}_s)$ is the largest number of edges incident to any single vertex in $\overline{\mathcal{G}}_s$. However, with given K and M_0, this bound is not always achievable. To see this, assume $\overline{\mathcal{G}}_s$ has $K-1$ edges. Connecting these edges to the same vertex yields an upper bound of $\Delta(\overline{\mathcal{G}}_s)+1=K$. But no matter how we allocate these edges, there are always unconnected vertices in $\overline{\mathcal{G}}_s$, which indicates that $\chi(\overline{\mathcal{G}}_s)<K$. Hence, the upper bound is not achievable here.

We are thus motivated to derive an achievable upper bound of U_s as a function of K and M_0. We first note that a set of pairwise unconnected vertices (a.k.a. an independent set) of $\mathcal{G}_s$, denoted by $\mathcal{V}_I$, must be transmitted separately because their corresponding packets all conflict with each other. The size of $\mathcal{V}_I$ is K when there is no edge in $\mathcal{G}_s$, indicating that $U_s=|\mathcal{V}_I|=K$. Then, whenever a new edge is added to the graph, we can maximize U_s by maximizing $|\mathcal{V}_I|$, which means that the edge should not connect two vertices in $\mathcal{V}_I$ whenever possible. Explicitly, our upper bound on U_s is derived iteratively:

- When there is no edge in the graph, we have $U_s=K$;
- When there are $M_0=[1,K-1]$ edges, we can use them to connect $\boldsymbol{v}_1$ with $\boldsymbol{v}_2\cdots\boldsymbol{v}_K$. Since $\boldsymbol{v}_2\cdots\boldsymbol{v}_K$ are independent, we have $U_s=K-1$;
- When there are up to $K-2$ additional edges, i.e, when $M_0=[K,2K-3]$, we can use these additional edges to connect $\boldsymbol{v}_2$ with $\boldsymbol{v}_3\cdots\boldsymbol{v}_K$. Since $\boldsymbol{v}_3\cdots\boldsymbol{v}_K$ are independent, we have $U_s=K-2$;
- The iterations will terminate when the graph is complete, i.e., when $M_0=K(K-1)/2$ and $U_s=1$.

It can be easily proved that any reallocation of edges will reduce the U_s derived above to a smaller value. Our upper bound on U_s has the following stair-case profile:

Property 3. *All S-IDNC graphs with K vertices and M_0 edges have:*

$$U_s\leqslant\begin{cases}K, & M_0=0\\ K-1, & M_0=[1,K-1]\\ K-2, & M_0=[K,2K-3]\\ \cdots, & \cdots\\ 1, & M_0=K(K-1)/2\end{cases}\tag{3}$$

2) U_s as a function of system parameters: In addition to finding U_s for a given S-IDNC graph, we are also interested in the statistical characteristics of U_s, for which we assume that $\boldsymbol{A}$ (and thus $\mathcal{G}_s$) is obtained as a consequence of random packet erasures in the systematic transmission phase.

For wireless broadcast, a common assumption on random packet erasures is that they are i.i.d. random variables with a probability of P_e. Under this assumption, a similar question has already been introduced and answered for the RLNC technique. It has been shown in [6], [36], [37] that the block completion time of RLNC scales as $\mathcal{O}(\ln(N))$ when K is a constant. Consequently, the throughput of RLNC vanishes with increasing number of receivers N. To prevent zero throughput, it has been proved in [38] that K should scale faster than $\ln(N)$.

Since the throughput of RLNC is optimal, it cannot be exceeded by the throughput of S-IDNC. Hence, we can infer that the throughput of S-IDNC should also follow a vanishing behavior with increasing N. However, its rate and specific dependence on system parameters have not been fully characterized in the literature. In this subsection, we answer this question through the following theorem:

Theorem 1. *The mean of the absolute minimum block completion time U_s is a function of the block size K, the number of receivers N, and packet erasure probability P_e as follows:*

$$E[U_s] = \left[-K\left(\frac{1}{2} + o(1)\right)\log_K(1 - P_e^2)\right] \cdot N \tag{4}$$

$$= c(K, P_e) \cdot N \tag{5}$$

where $o(1)$ is a small term that approaches zero with increasing K.

Proof: Our approach is to model the complementary S-IDNC graph $\overline{\mathcal{G}}_s$ as a random graph with i.i.d. edge generating probability. Recall that two vertices in $\overline{\mathcal{G}}_s$ are connected if the two data packets conflict, i.e., if at least one receiver has missed both packets. Therefore, the edge generating probability, denoted by P_c, is calculated as:

$$P_c = 1 - (1 - P_e^2)^N, \tag{6}$$

Then, the key is to prove that different edges are generated independently. We first consider the independency between two adjacent edges, say $\boldsymbol{e}_{1,2}$ and $\boldsymbol{e}_{1,3}$, which share $\boldsymbol{v}_1$. The information carried

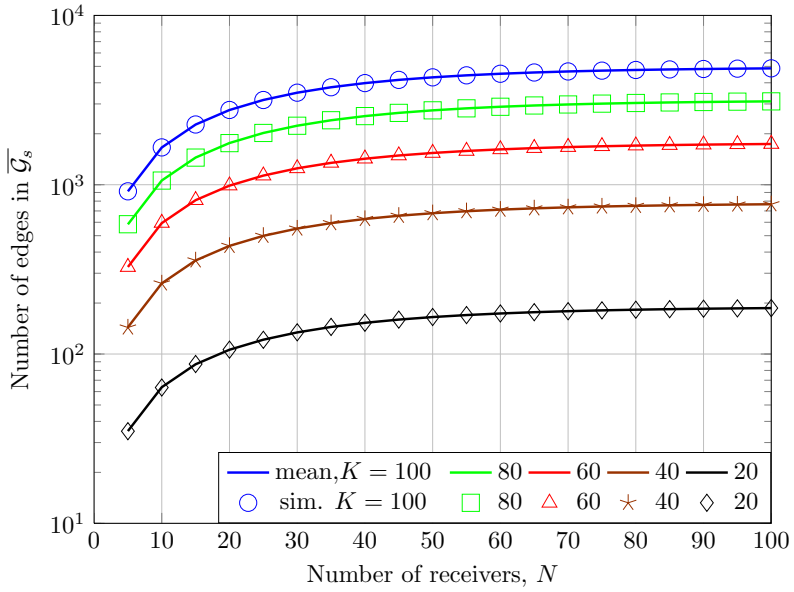


Fig. 2. The mean and simulated number of edges in $\overline{\mathcal{G}}_s$ when $P_e = 0.2$ and $K \in [20, 100]$.

by $\boldsymbol{e}_1$ (resp. $\boldsymbol{e}_2$) is that there is at least one receiver who wants both $\mathbf{p}_1$ and $\mathbf{p}_2$ (resp. $\mathbf{p}_1$ and $\mathbf{p}_3$). Hence, the mutual information between $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ is that $\mathbf{p}_1$ is wanted by at least one receiver, which happens with a probability of $1-(1-P_e)^N$. Thus:

$$I(\boldsymbol{e}_{1,2};\boldsymbol{e}_{1,3}) \leqslant H((1-P_e)^N, 1-(1-P_e)^N) \triangleq H(\mathbf{p}_1) \tag{7}$$

The inequality holds because other edges incident to $\mathbf{p}_1$ also contribute to $H(\mathbf{p}_1)$. It is clear that $I(\boldsymbol{e}_{1,2};\boldsymbol{e}_{1,3})$ quickly converges to zero with increasing N, indicating that $\boldsymbol{e}_{1,2}$ and $\boldsymbol{e}_{1,3}$ are asymptotically independent of each other. We note that two disjoint edges in $\overline{\mathcal{G}}_s$ share no mutual information, and thus are mutually independent. Therefore, we can assume that edges in $\overline{\mathcal{G}}_s$ are independently generated.

Consequently, $\overline{\mathcal{G}}_s$ can be modeled as an Erdõs-Rényi random graph [39], which has K vertices and i.i.d. edge generating probability of P_c. Fig. 2 compares the mean number of edges (with a value of $K(K-1)/2 \cdot P_c$) of our proposed random graph model and the simulated average number of edges in $\overline{\mathcal{G}}_s$. Our model shows virtually no deviation under all considered values of N and K. From graph theory, given K and P_c, almost every random graph $\overline{\mathcal{G}}_s$ has a chromatic number of:

$$\chi(\overline{\mathcal{G}}_s) = \frac{K}{\log K}\left(\frac{1}{2}+o(1)\right)\log\frac{1}{1-P_c} \tag{8}$$

Since $U_s = \chi(\overline{\mathcal{G}}_s)$, the above value is the mean of U_s. By substituting (6) into (8) we obtain (4). ■

Theorem 1 has the following important corollary:

Corollary 1. *The mean of the absolute minimum block completion time, $E[U_s]$, of S-IDNC increases linearly with the number of receivers in wireless broadcast with i.i.d. random packet erasures.*

Then, by noting that the mean block completion time of the coded transmission phase is lower bounded by $E[U_s]$, we conclude that the throughput of S-IDNC is significantly affected by the number of receivers. S-IDNC may not be a good choice when the number of receivers is large. We note that the above theorem and corollary are not directly applicable to G-IDNC, because the edge generating probability in G-IDNC is quite different. Interested readers are referred to [40] for more information.

B. Absolute minimum average packet decoding delay D_s

Unlike U_s, to the best of our knowledge there is no existing hardness result on finding D_s. In this subsection, we address it through the following theorem and then propose an upper bound on D_s.

Theorem 2. *It is NP-hard to find the absolute minimum APDD D_s of S-IDNC.*

Proof: In order to prove it, we introduce the concept of *perfect S-IDNC solution*:

Definition 3. *An S-IDNC solution is perfect and is denoted by $\mathcal{S}_p$ if every receiver R_n can decode one of its $|\mathcal{W}_n|$ wanted data packets from each of the first $|\mathcal{W}_n|$ coding sets in $\mathcal{S}_p$.*

This definition implies that $\mathcal{S}_p$, if it exists, offers the perfect packet decoding. Hence, its minimum APDD $D_{\mathcal{S}_p}$ is not only the absolute minimum APDD D_s of S-IDNC, but also a lower bound $\underline{D}$ of the minimum APDD of all other linear network coding techniques. Moreover, it can be easily shown that this lower bound can only be achieved by $\mathcal{S}_p$.

It has been proved in [41] that it is NP-hard to determine the achievability of $\underline{D}$. Hence, it is NP-hard to determine the existence of $\mathcal{S}_p$. Then by contradiction, if it is not NP-hard to find D_s, we can easily determine the existence of $\mathcal{S}_p$ by comparing D_s with $D_{\mathcal{S}_p}$. Therefore, it is NP-hard to find D_s. ■

Besides the NP-hardness, D_s has the following property:

Property 4. *The absolute minimum APDD D_s is upper bounded by U_s as*

$$D_s \leqslant \frac{U_s + 1}{2} \tag{9}$$

Proof: Given an S-IDNC solution $\mathcal{S}_s = \{\mathcal{M}_u\}_{u=1}^{U}$, let $T(u) = \sum_{\mathbf{p}_k \in \mathcal{M}_u} T_k$ be the number of receivers who can decode a data packet from $\mathcal{M}_u$. The minimum APDD of $\mathcal{S}_s$ is thus:

$$D_{\mathcal{S}_s} = \frac{1}{T} \sum_{u=1}^{U} T(u) \cdot u \tag{10}$$

which is maximized when $\{T(u)\}_{u=1}^{U} = \frac{T}{U}$. In this case, $D_{\mathcal{S}_s} = \frac{U+1}{2}$. Applying this result to an S-IDNC solution with absolute minimum block completion time $U = U_s$, we obtain the result. ■

Our proof indicates that, although it is NP-hard to achieve D_s, we can still effectively reduce APDD by reducing the S-IDNC solution size. Before we further explore this result to implement S-IDNC, we would like to compare the performance limits of S-IDNC that we have just derived with G-IDNC.

C. S-IDNC vs. G-IDNC

In this subsection, we address the question of *how does S-IDNC compare with G-IDNC?*

We first note that the NP-hardness of finding D_s also holds for D_g. This is because the perfect S-IDNC solution $\mathcal{S}_p$ is also the best possible G-IDNC solution. For the throughput, we first present a relation between S- and G-IDNC graphs (proved in the appendix):

Theorem 3. *The minimum clique partition solutions of S-IDNC and G-IDNC graphs have the same size. In other words,* $\chi(\overline{\mathcal{G}}_s) = \chi(\overline{\mathcal{G}}_g)$.

However, the above theorem does not imply $U_s = U_g$. This is because G-IDNC does not have Property 1. Explicitly, by removing a vertex from $\mathcal{G}_g$, more edges and larger cliques may be generated, and thus the absolute minimum block completion time U_g can be smaller than $\chi(\overline{\mathcal{G}}_g)$ of the original G-IDNC graph $\mathcal{G}_g$ [25]. We thus have $U_g \leqslant U_s$. We note, however, that a systematic way of finding U_g other than brute-force search remains widely open.

Therefore, when there are no packet erasures, the throughput of G-IDNC is at least as good as S-IDNC. But is this still true in more realistic erasure-prone scenarios? In the next section, we will design S-IDNC transmission schemes under packet erasures and compare them with G-IDNC. We will apply the above theorem to show that G-IDNC cannot outperform S-IDNC under certain circumstances.

IV. S-IDNC TRANSMISSION SCHEMES

In this section, we design S-IDNC transmission schemes to compensate for packet erasures in the coded transmission phase, which are i.i.d. with a probability of P_e. To this end, the sender has to regularly collect feedback from the receivers about their packet reception state to make online coding decisions. We consider two common types of feedback frequency, namely:

1) fully-online feedback: feedback is collected after every coded transmission. However, this could be costly in wireless communications. We thus also consider a reduced feedback frequency next;
2) semi-online feedback: feedback is only collected after several (to be quantified later) coded transmissions;

To be able to design S-IDNC transmission schemes, two questions need to be answered first:

1) What is the optimization objective for throughput and decoding delay improvement?
2) What does the sender need to send to achieve it?

Before addressing these questions, we first highlight some challenges:

Remark 1. *Under random packet erasures, a reasonable measure of throughput is the mean block completion time $E[U_T]$ of the coded transmission phase. However, it is intractable to minimize $E[U_T]$. To see this, let us consider the stochastic shortest path (SSP) method [24]. In SSP method, the state space comprises the current SFM and its successors, and thus has a prohibitively large size with a value of 2^T, where T is the number of "1"s in $\boldsymbol{A}$. The action space for each state comprises all cliques/coding sets, which is NP-hard to find [42]. Then, $E[U_T]$ is recursively minimized by examining all the states and the associated actions. Such examination is necessary, because the packet erasures can take any pattern and are not predictable. But it makes $E[U_T]$ intractable to minimize. To overcome this difficulty, we will turn to optimization objectives that are heuristic, but still based on SSP optimization principles.*

Remark 2. *It is intractable to minimize the APDD D_T of the coded transmission phase due to the NP-hardness of finding D_s, because otherwise by setting $P_e = 0$, the minimum D_T is equal to D_s. To overcome this difficulty, we will give higher priority to the minimization of block completion time. In other words, we first minimize the block completion time. Then among the resultant coding decisions,*

we choose the one that minimizes the decoding delay. Our prioritization reflects the motivation of using network coding, that is, to achieve better throughput performance. It also provides bounded decoding delay performance as we have shown in (9)*. This will also be confirmed by our simulations, which show that D_T generally decreases with decreasing U_T.*

A. Fully-online Transmission Scheme

With fully-online feedback frequency, the sender only transmits one coded packet before collecting feedback. Under the SSP method, the current state is the current SFM $\boldsymbol{A}$, the absorbing state is the all-zero SFM and is denoted by $\boldsymbol{A}_0$. The action space comprises all the S-IDNC coding sets of $\boldsymbol{A}$. The cost of each action is one, for it consumes one transmission. The block completion time U_T is thus equal to the number of transitions (a.k.a. path length or distance) between $\boldsymbol{A}$ and $\boldsymbol{A}_0$.

According to Remark 1, it is intractable to choose an action/coded packet that minimizes the expected path length (and thus $E[U_T]$). As a heuristic alternative, we propose to choose an action/coded packet that belongs to the shortest path from $\boldsymbol{A}$ to $\boldsymbol{A}_0$, which has a length of U_s. This choice guarantees that, upon the reception of the coded packet at all interested receivers, the shortest distance between the updated state $\boldsymbol{A}'$ and $\boldsymbol{A}_0$ is minimized to $U_s - 1$. To this end, the coded packet must belong to a minimum clique partition solution $\mathcal{S}_c$. Otherwise, the shortest distance between $\boldsymbol{A}'$ and $\boldsymbol{A}_0$ is still U_s.

We then reduce APDD by forcing the coded packet to be maximal (and thus serving the maximal number of receivers). However, cliques in a minimum clique partition solution are not necessarily maximal. Hence, we further require the coded packet to belong to a set of U_s maximal cliques that together cover all the data packets. This set is also an S-IDNC solution and is denoted by $\mathcal{S}_m$.

In conclusion, we propose the following coded packet $\mathcal{M}_f$ for fully-online transmission scheme:

Given an SFM instance, the preferred coded packet $\mathcal{M}_f$ is the most wanted coded packet in $\mathcal{S}_m$, where $\mathcal{S}_m$ is an S-IDNC solution that contains U_s maximal cliques.

B. Semi-online Transmission Scheme

The fully-online transmission scheme is costly, not only in collecting feedback, but also in computational load, as it has to find $\mathcal{S}_m$ in every time slot. These problems can be mitigated by partitioning the

coded transmission phase into rounds. In each round, the sender transmits a complete S-IDNC solution and only collects feedback at the end of each round. We call this scheme the semi-online scheme.

Under the SSP method, the action space is the set of all S-IDNC solutions $\mathbb{S}_s$, and the cost of each action is the solution size $|\mathcal{S}_s|$, which is equal to the length of a semi-online transmission round. The total cost is thus equal to the block completion time.

According to Remark 1, it is intractable to minimize the expected total cost (and thus $E[U_T]$). As a heuristic alternative, we propose to minimize the expected cost of the shortest path between $\boldsymbol{A}$ and $\boldsymbol{A}_0$. The shortest path has a length of one, representing the event that every coded packet of the chosen solution $\mathcal{S}_s$ is received by all the interested receivers after only one semi-online round. Denote the probability of this event by P_s. Then the expected cost is $|\mathcal{S}_s|/P_s$, where P_s is calculated as:

$$P_s = \prod_{k=1}^{K} (1 - P_e^{d_k})^{T_k} \tag{11}$$

Here d_k is called the packet diversity and is defined below.

Definition 4. *The diversity d_k of data packet $\mathbf{p}_k$ is the number of coding sets in $\mathcal{S}_s$ that comprise $\mathbf{p}_k$.*

We note that the minimum clique partition solution $\mathcal{S}_c$ is not a preferred semi-online S-IDNC solution. Although $\mathcal{S}_c$ offers the smallest solution size ($|\mathcal{S}_c| = U_s$), it does not maximize P_s because every data packet has a diversity of only one due to disjoint cliques in $\mathcal{S}_c$. Instead, the $\mathcal{S}_m$ we have proposed for the fully-online case can offer a higher P_s than $\mathcal{S}_c$ due to possibly overlapping maximal cliques, while also offering the smallest solution size.

We still wish to answer the following question before choosing $\mathcal{S}_m$ as our preferred semi-online S-IDNC solution: *Is there a solution that, though large in its size, provides higher packet diversities, so that P_s is maximized?*

An explicit answer to this question is difficult to obtain, because it requires the examination of all the solutions of size greater than U_s. Such search is costly and does not provide any insight into this question. Moreover, a solution with a larger block completion time is unlikely to provide higher packet diversities due to the following property of S-IDNC solutions:

Property 5. *Every coding set in an S-IDNC solution comprises at least one data packet with a diversity*

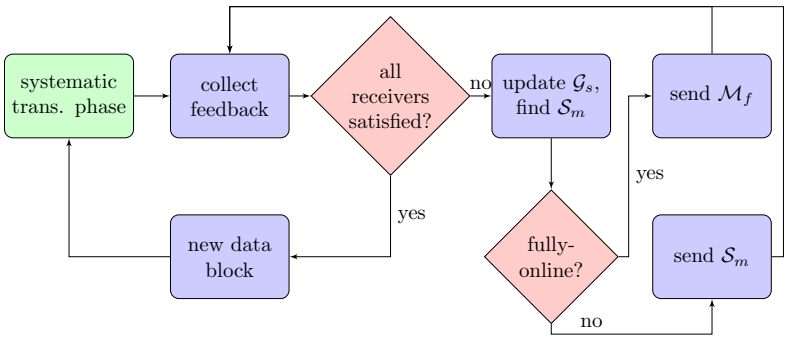


Fig. 3. The fully- and semi-online transmission schemes.

of one.

This property holds because if every data packet in a coding set has a diversity of greater than one, then this coding set can be removed from the solution without affecting the completeness of the solution. Due to the above property, an S-IDNC solution $\mathcal{S}_s$ has at least $|\mathcal{S}_s|$ data packets with a diversity of only one. According to (11), these unit-diversity data packets reduce P_s the most.

Therefore, we choose $\mathcal{S}_m$ for throughput improvement. Then, by taking into account our secondary optimization objective, i.e., the APDD, we define our preferred semi-online S-IDNC solution as follows:

Given an SFM instance, the proposed semi-online S-IDNC solution is $\mathcal{S}_m$, which comprises a set of U_s maximal cliques. The cliques are sorted for transmission in the descending order of their numbers of targeted receivers to minimize the APDD.

A flow-chart of the proposed two transmission schemes are presented in Fig. 3. Both the fully- and semi-online IDNC schemes require finding $\mathcal{S}_m$. Since packet diversity is not a concern in graph theory, there is no algorithms to find $\mathcal{S}_m$ in the graph theory literature. Hence, we will design algorithms dedicated for S-IDNC in the next section. Before doing so, however, we briefly compare S-IDNC and G-IDNC under the above two transmission schemes.

C. S-IDNC vs. G-IDNC

With fully-online feedback, the sender can update the G-IDNC graph and add new edges representing coding opportunities after every transmission. The throughput of G-IDNC is thus better than S-IDNC. But the price is high computational load, because G-IDNC graph is much larger than S-IDNC graph

($\mathcal{O}(NK)$ v.s. K). However, during a semi-online transmission round, the sender cannot update SFM due to the absence of receiver feedback. Consequently, it does not update the G-IDNC graph $\mathcal{G}_g$ [29], and only sends the minimum clique partition solution of $\mathcal{G}_g$, which, according to Theorem 3, has the same size as the minimum clique partition solution of S-IDNC. We thus have the following corollary:

Corollary 2. *G-IDNC cannot reduce the length of a semi-online transmission round compared to S-IDNC.*

V. S-IDNC Coding Algorithms

The two transmission schemes we proposed in the last section require finding $\mathcal{S}_m$, an S-IDNC solution that contains U_s maximal coding sets. In this section, we develop its optimal and heuristic algorithms.

A. Optimal S-IDNC coding Algorithm

Our optimal S-IDNC coding algorithm finds $\mathcal{S}_m$ in two steps:

Step-1 *Find all the maximal coding sets (maximal cliques):* This problem is NP-hard in graph theory. We apply an exponential algorithm, called Bron-Kerbosch (B-K) algorithm [42]. The group of all maximal cliques is denoted by $\mathcal{A}$.

Step-2 *Find $\mathcal{S}_m$ from $\mathcal{A}$:* We propose a branching algorithm in Algorithm 1. The intuition behind this algorithm is that, if a data packet $\mathbf{p}_k$ belongs to d_k maximal coding sets in $\mathcal{A}$, then one of these d_k maximal coding sets must be included in $\mathcal{S}_m$ for the completeness of $\mathcal{S}_m$. In the extreme case where $d_k = 1$, the sole maximal coding set that contains $\mathbf{p}_k$ must be included in $\mathcal{S}_m$. Below is an example of Algorithm 1.

Example 4. *Consider the graph model in Fig. 4. In Step-1, we find all the maximal cliques:* $\mathcal{A} = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5), (\mathbf{p}_3, \mathbf{p}_4), (\mathbf{p}_4, \mathbf{p}_6), (\mathbf{p}_5, \mathbf{p}_6)\}$. *Then in Step-2:*

1) *Initially,* $\mathcal{S} = \emptyset$, $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S} = \mathcal{A}$, *and the set of data packets not included in* $\mathcal{S}$ *is* $\overline{\mathcal{P}} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6\}$. *Since* $\mathbf{p}_1$ *is only included in* $(\mathbf{p}_1, \mathbf{p}_3)$ *and* $\mathbf{p}_2$ *is only included in* $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)$, *these two coding sets must be added to* $\mathcal{S}$. *Hence,* $\mathcal{S} = \{(\mathbf{p}_1, \mathbf{p}_3), (\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5)\}$ *after the first two iterations;*

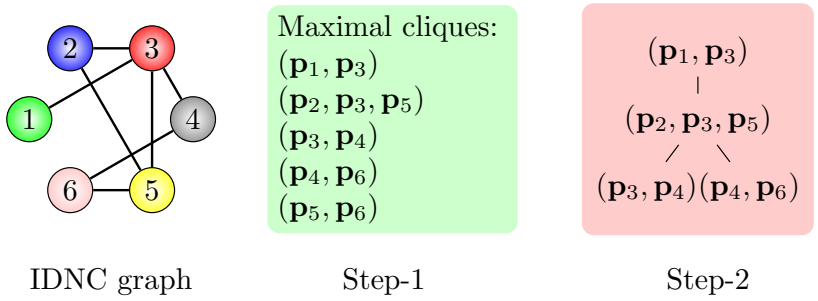


Fig. 4. An example of the 2-step optimal S-IDNC coding algorithm.

2) *The set of data packets not included in* $\mathcal{S}$ *is* $\overline{\mathcal{P}} = \{\mathbf{p}_4, \mathbf{p}_6\}$*, and the remaining maximal coding sets are* $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S}$=$\{(\mathbf{p}_3, \mathbf{p}_4),\ (\mathbf{p}_4, \mathbf{p}_6),\ (\mathbf{p}_5, \mathbf{p}_6)\}\}$*. Since* $\mathbf{p}_4$ *has a diversity of 2 under* $\overline{\mathcal{S}}$ *due to* $(\mathbf{p}_3, \mathbf{p}_4)$ *and* $(\mathbf{p}_4, \mathbf{p}_6)$*, we branch* $\mathcal{S}$ *into two successors:* $\mathcal{S}_1 = \{(\mathbf{p}_1, \mathbf{p}_3),(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5),(\mathbf{p}_4, \mathbf{p}_5)\}$ *and* $\mathcal{S}_2 = \{(\mathbf{p}_1, \mathbf{p}_3),(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5),(\mathbf{p}_4, \mathbf{p}_6)\}$*. Since* $\mathcal{S}_2$ *contains all data packets and there are no other branching opportunities, the algorithm stops and outputs* $\mathcal{S}_2$ *as* $\mathcal{S}_m$*.*

B-K algorithm and Algorithm 1 constitute our optimal S-IDNC coding algorithm. It produces all the valid $\mathcal{S}_m$. Among these solutions, we can choose the one that optimizes a secondary criteria, such as the one offering the smallest $D_{\mathcal{S}}$, or the largest P_S, calculated using (11).

B. Hybrid S-IDNC Coding Algorithm

Algorithm 1 is memory demanding, because the number of candidate solutions grows exponentially with branching. Thus, we propose a heuristic alternative to it. The idea is to iteratively maximize the number of data packets included in $\mathcal{S}_m$. The algorithm is given in Algorithm 2.

B-K algorithm and Algorithm 2 constitute our hybrid S-IDNC coding algorithm. It produces only one S-IDNC solution, with no guarantee on the solution size. It is still computational expensive due to B-K algorithm. Thus, we develop a polynomial time heuristic S-IDNC coding algorithm next.

C. Heuristic S-IDNC coding Algorithm

Algorithm 3 is a simple algorithm that heuristically finds the maximum (the largest maximal) clique of a graph. The intuition behind this algorithm is that, a vertex is very likely to be in the maximum clique if it is incident by the largest number of edges. Variations of this algorithm have been developed

Algorithm 1 Optimal S-IDNC solution search

input: the group of all maximal coding sets, $\mathcal{A}$;

initialization: a set $\mathcal{B}$ of solutions, $\mathcal{B}$ only contains an empty solution $\mathcal{S} = \emptyset$. A counter $u = 1$;

while no solution in $\mathcal{B}$ contains all data packets, **do**

  while there is a solution in $\mathcal{B}$ with size $u - 1$, **do**

    Denote this solution by $\mathcal{S} = \{\mathcal{M}_1, \cdots, \mathcal{M}_{u-1}\}$. Denote the data packets included in $\mathcal{S}$ by $\mathcal{P} = \bigcup_{i=1}^{u-1}\{\mathcal{M}_i\}$ and all data packets not included in $\mathcal{S}$ by $\overline{\mathcal{P}} = \mathcal{P}_K \setminus \mathcal{P}$. Also denote the maximal coding sets not included in $\mathcal{S}$ by $\overline{\mathcal{S}} = \mathcal{A} \setminus \mathcal{S}$;

    Pick from $\overline{\mathcal{P}}$ the data packet $\mathbf{p}$ that has the smallest diversity d under $\overline{\mathcal{S}}$. Denote the d coding sets which contain $\mathbf{p}$ by $\mathcal{M}'_1, \cdots, \mathcal{M}'_d$;

    Branch $\mathcal{S}$ into d new solutions, $\mathcal{S}'_1, \cdots, \mathcal{S}'_d$. Then, add $\mathcal{M}'_1, \cdots, \mathcal{M}'_d$ to these solutions, respectively. The sizes of the new solutions are u;

  end while

  $u = u + 1$;

end while

Output the solutions in $\mathcal{B}$ that contain all data packets.

in the literature [12], [13], [24]. But this algorithm has not been applied to finding a complete S-IDNC solution, and its computational complexity has not been identified yet.

The computational complexity of Algorithm 3 is polynomial in the number of data packets K. The highest computational cost occurs when the input graph is complete, i.e., when all vertices are connected to each other. In this case, only one vertex will be removed in each iteration. Thus, the number of remaining vertices in iteration-i will be $K - i$, $\forall i \in [0, K-1]$. Then, to find the vertex with the largest number of incident edges, we need $K - i$ comparisons. The total computational cost is thus in the order of $\sum_{i=0}^{K-1} K - i = K(K-1)/2$. Hence, the computational complexity of Algorithm 3 is at most $\mathcal{O}(K^2)$.

We apply Algorithm 3 to iteratively find $\mathcal{S}_m$ in Algorithm 4. In each iteration, we find a clique using Algorithm 3, maximize it by adding more vertices to it whenever possible, and then remove it from the S-IDNC graph. This will increase the diversities of the added vertices/packets. Below is an example:

Algorithm 2 Hybrid S-IDNC solution search

input: the group of all maximal coding sets, $\mathcal{A}$;

initialization: an empty solution $\mathcal{S} = \emptyset$, a counter $u = 1$, packet set $\mathcal{P} = \mathcal{P}_K$;

while $\mathcal{S}$ does not contain all data packets, **do**

    find the coding set $\mathcal{M}$ in $\mathcal{A}$ that contains the largest number of data packets in $\mathcal{P}$;

    Add $\mathcal{M}$ to $\mathcal{S}$ and remove data packets in $\mathcal{M}$ from $\mathcal{P}$;

    $u = u + 1$;

end while

Output the solution $\mathcal{S}$.

Algorithm 3 Heuristic maximum clique search

input: graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$;

initialization: an empty vertex set $\mathcal{V}_{\text{keep}}$;

while $\mathcal{G}$ is not empty **do**

    add to $\mathcal{V}_{\text{keep}}$ the vertex $\boldsymbol{v}$ which has the largest number of edges incident to it;

    update $\mathcal{G}$ by deleting $\boldsymbol{v}$ and all the vertices not connected to $\boldsymbol{v}$ (*These vertices can be ignored because they cannot be part of the target clique, which contains* $\boldsymbol{v}$);

end while

vertices in $\mathcal{V}_{\text{keep}}$ are pair-wise connected, and no other vertices can be added to them. Hence, $\mathcal{V}_{\text{keep}}$ is a maximal clique.

Example 5. *Consider the graph $\mathcal{G}_s$ in Fig. 1(b). In the first two iterations, the algorithm will choose $\mathcal{M}_1 = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$ and $\mathcal{M}_2 = (\mathbf{p}_3, \mathbf{p}_6)$, respectively. In the third iteration, $\mathcal{V}_{covered} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_6\}$ and the algorithm can only choose $\mathcal{M}_3 = (\mathbf{p}_5)$. Among all the data packets in $\mathcal{V}_{covered}$, $\mathbf{p}_2$ can be added to $\mathcal{M}_3$. Thus $\mathcal{M}_3 = \{\mathbf{p}_2, \mathbf{p}_5\}$. The algorithm then stops and outputs $\mathcal{S}_m = \{(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4), (\mathbf{p}_3, \mathbf{p}_6), (\mathbf{p}_2, \mathbf{p}_5)\}$.*

In conclusion, we proposed an optimal algorithm that finds $\mathcal{S}_m$, as well as its hybrid and heuristic alternatives. The output $\mathcal{S}_m$ is used as the S-IDNC solution for the semi-online transmission scheme. If fully-online transmission scheme is applied, the transmitted coding set $\mathcal{M}_f$ is chosen from $\mathcal{S}_m$.

Algorithm 4 Heuristic S-IDNC solution search

input: a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$;

initialization: an empty vertex set $\mathcal{V}_{\text{covered}}$, a working graph $\mathcal{G}_w = \mathcal{G}$, and a counter $i = 0$;

while $\mathcal{V}_{\text{covered}} \neq \mathcal{V}$ **do**

  Find the maximum clique in $\mathcal{G}_w$ using Algorithm 3. Denote it by $\mathcal{M}_i$;

  Find the vertices in $\mathcal{V}_{\text{covered}}$ which are connected to $\mathcal{M}_i$. Denote their set by $\mathcal{V}_i$ (*They are the candidate vertices that could be added to* $\mathcal{M}_i$.);

  Generate a subgraph of $\mathcal{G}$ whose vertex set is $\mathcal{V}_i$. Denoted this subgraph by $\mathcal{G}_i'(\mathcal{V}_i, \mathcal{E}_i)$;

  Find the maximum clique in $\mathcal{G}_i'$ using Algorithm 1, denoted it by $\mathcal{M}_i'$ (*All vertices in* $\mathcal{M}_i'$ *are connected to each other and thus can all be added to* $\mathcal{M}_i$.);

  Update $\mathcal{V}_{\text{covered}}$ by adding vertices in $\mathcal{M}_i$ into it;

  Update $\mathcal{G}_w$ by removing $\mathcal{M}_i$ from it;

  Update $\mathcal{M}_i$ as $\mathcal{M}_i = \mathcal{M}_i \cup \mathcal{M}_i'$ (*The new clique is at least as large as the old one, and thus provides higher packet diversity*);

  $i = i + 1$;

end while

VI. Simulations

In this section, we present the simulated throughput and decoding delay performance of S-IDNC (abbreviated as S- in the figures) under different scenarios, including under full- and semi-online transmission schemes, and under the use of optimal, hybrid, and heuristic coding algorithms (abbreviated in the figures as Opt., Hybr., and Heur., respectively). The packet block size is $K = 15$. The number of receivers N is chosen between 5 and 40. The packet erasures are i.i.d. among the channels between the sender and the receivers probability with a probability of $P_e = 0.2$.

We also compare S-IDNC with RLNC and G-IDNC. For RLNC, we assume a sufficiently large finite field, so that its throughput is almost surely optimal and serves as a benchmark. For G-IDNC, although its best performance is at least as good as S-IDNC (as we have explained in Section III-C), this advantage will not necessarily be reflected in our simulation results. This is because there has not been

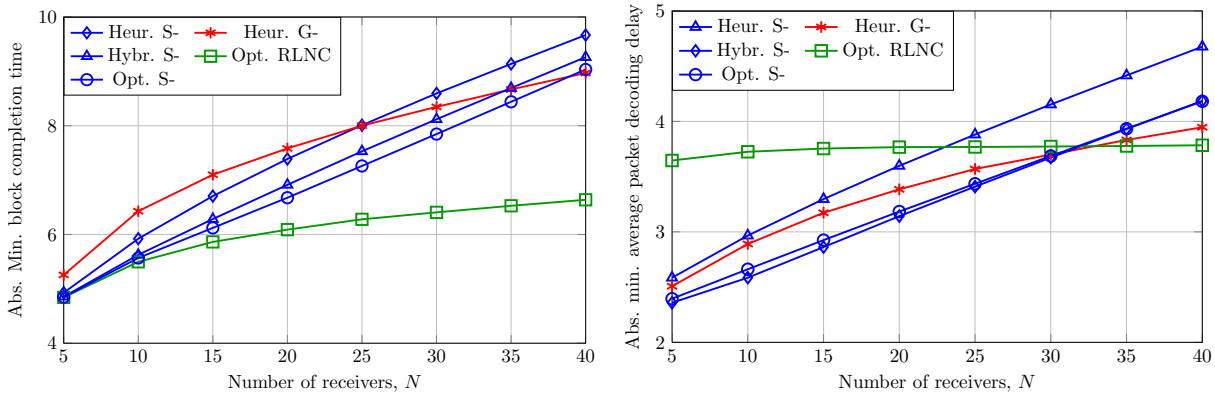
any optimal G-IDNC algorithm. Instead, we apply a heuristic algorithm (abbreviated as Heur. G- in the figures) proposed in [24], which aims at minimizing the block completion time. This aim coincides with our optimization priorities for S-IDNC in Remark 2, namely, to minimize the block completion time first.

We conduct three sets of simulations. The first set compares the performance limits of the three techniques. The results are presented in Fig. 5. Here for RLNC, its absolute minimum block completion time is equal to the size of the largest Wants set of the receivers. This number cannot be further reduced by any means, because otherwise the most demanding receivers cannot decode all its wanted data packets. The second (resp. third) set of simulations compares the throughput and decoding delay performance under fully-online (resp. semi-online) transmission scheme. The results are presented in Fig. 6 (resp. Fig. 7). We note that the performance of RLNC is the same under both schemes, because RLNC is feedback-free.

Our observations on S-IDNC are as follows:

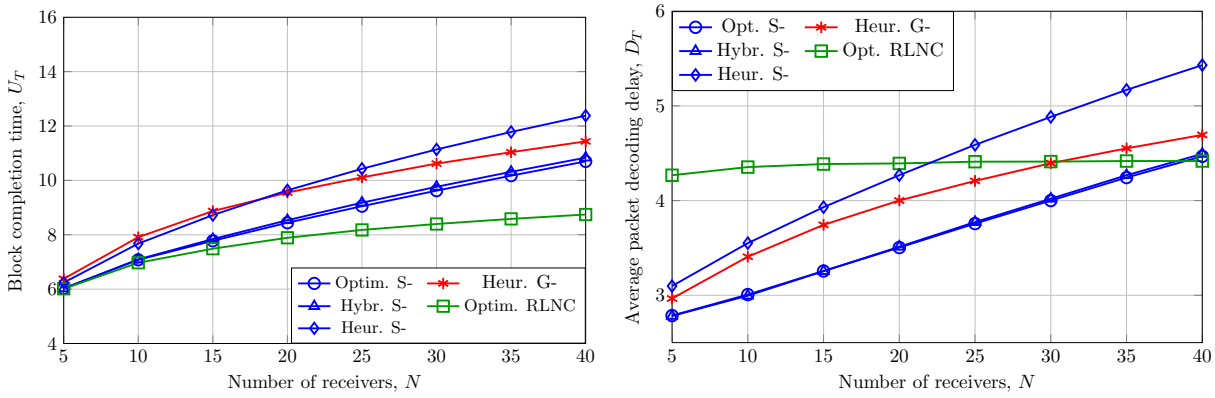
- The absolute minimum block completion time of S-IDNC increases almost linearly with N. This result matches Corollary 1;
- The fully-online transmission scheme always provides better throughput and decoding delay performance than the semi-online one;
- The optimal coding algorithm always provides better throughput performance than its hybrid and heuristic alternatives. This result verifies our choice of $\mathcal{S}_m$ for throughput improvement, because only the optimal coding algorithm can always produce $\mathcal{S}_m$, which has $|\mathcal{S}_m| = U_s$;
- However, the optimal coding algorithm does not necessarily minimize the APDD. For example, in Fig. 6(b), the hybrid algorithm provides smaller APDD than the optimal one under the fully-online transmission scheme;
- The performance gap between the optimal and hybrid algorithms is always marginal, and is much smaller than their gap with the heuristic one. Hence, the hybrid algorithm strikes a good balance between performance and computational load.

A cross comparison of RLNC, S-, and G-IDNC shows that:



(a) Block completion time (b) Average packet decoding delay

Fig. 5. The throughput and decoding delay performance limits of S- and G-IDNC, as well as RLNC.

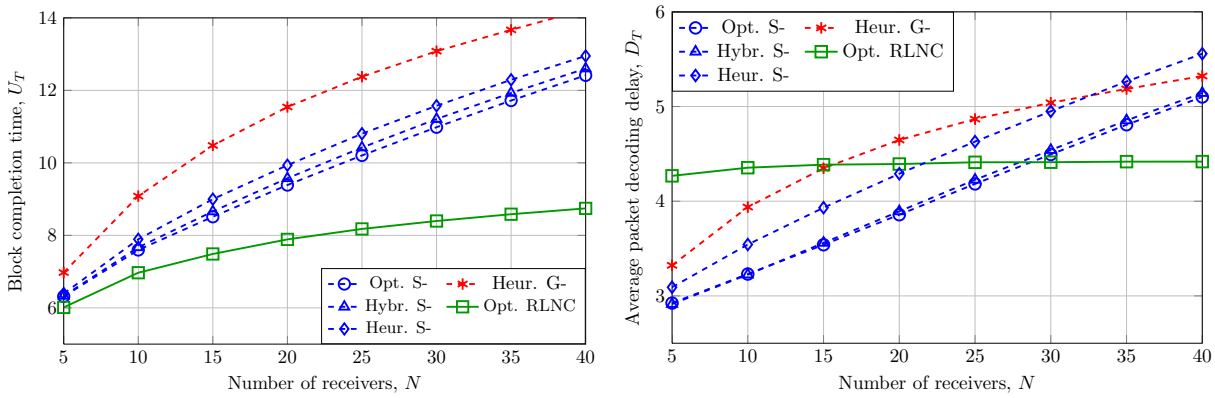


(a) Block completion time (b) Average packet decoding delay

Fig. 6. The throughput and decoding delay performance of fully-online transmission scheme when different coding algorithms are applied. It is compared with the performance of heuristic fully-online G-IDNC and RLNC.

- The throughput of RLNC is always the best. The throughput of S-IDNC is very close to RLNC when the number of receivers is small. Their gap increases with N;
- In general, the APDD of both S- and G-IDNC is better than RLNC. This advantage only vanishes when the block completion time of S- and G-IDNC becomes much larger than RLNC, which takes place when N is much larger than K;
- There is no clear winner between the performance of heuristic G-IDNC and optimal S-IDNC. We can expect that G-IDNC will outperform S-IDNC if its optimal coding algorithm is developed.

In summary, our simulations verified our theorems, propositions, and algorithms. They also demonstrated that, if we are concerned with both throughput and decoding delay performance, S-IDNC is a good alternative to RLNC when the number of receivers is not too large.



(a) Block completion time

(b) Average packet decoding delay

Fig. 7. The throughput and decoding delay performance of semi-online transmission scheme when different coding algorithms are applied. It is compared with the performance of heuristic semi-online G-IDNC and RLNC.

VII. Conclusion

In this paper, we studied the throughput and decoding delay performance of S-IDNC in broadcasting a block of data packets to wireless receivers under packet erasures. By using a random graph model, we showed that the throughput of S-IDNC decreases linearly with increasing number of receivers. By introducing the concept of perfect S-IDNC solution, we proved the NP-hardness of minimizing the average packet decoding delay. We also proposed two upper bounds on the throughput and decoding delay limits of S-IDNC. We considered two transmission schemes that requires fully- and semi- online feedback frequencies, respectively. By applying stochastic shortest path method, we showed that it is intractable to make optimal coding decisions in the presence of random packet erasures. We then used heuristic objective functions to determine the preferred coded packet(s) to send. We then developed the optimal S-IDNC coding algorithm and its complexity-reduced heuristics. We also compared S-IDNC with G-IDNC by proving the equivalence between the chromatic number of the complementary S-IDNC and G-IDNC graphs. We used this equivalence to show that G-IDNC can outperform S-IDNC when there are not packet erasures, but this is not always true when there are packet erasures.

Our work provides news understandings of S-IDNC. It will facilitate the extension of S-IDNC to applications in other network settings, such as cooperative data exchange and distributed data storage. We are also interested in designing approximation and heuristic algorithms for decoding delay minimization.

APPENDIX A
PROOF OF THEOREM 3

Theorem 3 requires the proof of $\chi(\overline{\mathcal{G}}_s) = \chi(\overline{\mathcal{G}}_g)$. Since every S-IDNC solution is also a G-IDNC solution, but a G-IDNC solution is not necessarily an S-IDNC solution, we have $U_s \geqslant U_g$, and thus $\chi(\overline{\mathcal{G}}_s) \geqslant \chi(\overline{\mathcal{G}}_g)$. Hence, here we only need to prove that $\chi(\overline{\mathcal{G}}_s) \leqslant \chi(\overline{\mathcal{G}}_g)$.

We first introduce the concept of *affiliated* S-IDNC graph $\mathcal{G}_{as}$ of a G-IDNC graph $\mathcal{G}_g$, which is construct as follows. Given $\mathcal{G}_g$ that involves K data packets and N receivers, we generate a graph $\mathcal{G}_{as}$ with K vertices, each representing a data packet. We then connect $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ in $\mathcal{G}_{as}$ if for every pair of $\{m, n\} \in [1, N]$, $\boldsymbol{v}_{i,m}$ and $\boldsymbol{v}_{j,n}$ are connected upon their existence in $\mathcal{G}_g$. In other words, we claim that $\mathbf{p}_i$ and $\mathbf{p}_j$ do not conflict if every vertex that represents $\mathbf{p}_i$ in $\mathcal{G}_g$ is connected to every vertex that represents $\mathbf{p}_j$ in $\mathcal{G}_g$.

Given an SFM $\boldsymbol{A}$, we can easily show that its S-IDNC graph $\mathcal{G}_s$ is the same as the affiliated S-IDNC graph $\mathcal{G}_{as}$ of its G-IDNC graph $\mathcal{G}_g$. Hence, our task becomes to prove that $\chi(\overline{\mathcal{G}}_{as}) \leqslant \chi(\overline{\mathcal{G}}_g)$, where $\chi(\overline{\mathcal{G}}_{as}) = U_s$. This statement is true if the following property is true:

Property 6. *After removing any clique $\mathcal{M}_g$ from $\mathcal{G}_g$, the chromatic number of the affiliated S-IDNC graph $\mathcal{G}_{as}$ is reduced by at most one.*

Since $\mathcal{G}_g$ is nonempty as long as $\mathcal{G}_{as}$ is nonempty, this property indicates that any clique partition solution of $\mathcal{G}_g$ must have a size of at least $\chi(\overline{\mathcal{G}}_{as})$, which will prove that $\chi(\overline{\mathcal{G}}_{as}) \leqslant \chi(\overline{\mathcal{G}}_g)$. Property 6 can be proved through induction:

1) If $\mathcal{M}_g$ does not contain any conflicting data packets in $\mathcal{G}_{as}$, then $\chi(\overline{\mathcal{G}}_{as})$ is reduced by at most one;
2) If $\mathcal{M}_g$ contains one pair of conflicting data packets in $\mathcal{G}_{as}$, then $\chi(\overline{\mathcal{G}}_s)$ is reduced by at most one;
3) If $\mathcal{M}_g$ already contains m pairs of conflicting data packets in $\mathcal{G}_{as}$, then modifying $\mathcal{M}_g$ to contain one more pair of conflicting data packets in $\mathcal{G}_{as}$ cannot further reduce $\chi(\overline{\mathcal{G}}_{as})$.

The first statement is self-evident, because the set of data packets included in such $\mathcal{M}_g$ is a clique of $\mathcal{G}_{as}$. By removing it, $\chi(\overline{\mathcal{G}}_{as})$ can be reduced by at most one.

To prove the second statement, without loss of generality we assume that the pair of conflicting data

packets is $(\mathbf{p}_1, \mathbf{p}_2)$. Then the set of data packets included in $\mathcal{M}_g$ takes a form of $\{\mathcal{M}_s, \mathbf{p}_1, \mathbf{p}_2\}$, where $\mathcal{M}_s$ is the set of pair-wise non-conflicting data packets, and thus is a clique of $\mathcal{G}_{as}$. Since $\mathbf{p}_1$ conflicts with $\mathbf{p}_2$, there exists at least one pair of unconnected vertices in $\mathcal{G}_g$ that represent $\mathbf{p}_1$ and $\mathbf{p}_2$. This pair is not included in $\mathcal{M}_g$, and thus is kept after removing $\mathcal{M}_g$ from $\mathcal{G}_g$. Hence, in the updated affiliated S-IDNC graph $\mathcal{G}'_{as}$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ exist, and are unconnected. Let the chromatic number of $\mathcal{G}'_{as}$ be U', then the minimum clique partition of $\mathcal{G}'_{as}$ takes a form of $\{\mathcal{M}_1, \cdots, \mathcal{M}_{U'}\}$, which keeps $\mathbf{p}_1$ and $\mathbf{p}_2$ in different coding sets. Then, since $\mathcal{M}_s$ is a clique of $\mathcal{G}_{as}$, $\{\mathcal{M}_s, \mathcal{M}_1, \cdots, \mathcal{M}_{U'}\}$ is a partition of $\mathcal{G}_{as}$ with a size of $U'+1$. Thus, $U' \geqslant \chi(\overline{\mathcal{G}}_{as}) - 1$, implying that $\chi(\overline{\mathcal{G}}_s)$ is reduced by at most one after removing $\mathcal{M}_g$ from $\mathcal{G}_g$.

The proof of the third statement is similar to the second one, and thus is omitted here. According to the above three statements, no matter how many conflicting data packets are included in $\mathcal{M}_g$, after removing $\mathcal{M}_g$ from $\mathcal{G}_g$, the chromatic number of the affiliated S-IDNC graph $\mathcal{G}_{as}$ is reduced by at most one. Therefore, $\chi(\overline{\mathcal{G}}_g) \geqslant \chi(\overline{\mathcal{G}}_{as})$. Since $\mathcal{G}_{as}$ is the same as $\mathcal{G}_s$, we have $\chi(\overline{\mathcal{G}}_g) \geqslant \chi(\overline{\mathcal{G}}_s)$ and Theorem 3 is proved.

REFERENCES

[1] J. K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2008.

[2] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, 2000.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, 2008.

[4] C. Fragouli, J. Widmer, and J. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, 2008.

[5] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *Proc. NetCod*, 2008.

[6] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5511–5524, Dec. 2008.

[7] T. Tran, T. Nguyen, and B. Bose, "A joint network-channel coding technique for single-hop wireless networks," in *NetCod*, 2008.

[8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914–925, Feb. 2009.

[9] D. E. Lucani, M. Médard, and M. Stojanovic, "Broadcasting in time-division duplexing: A random linear network coding approach," in *Proc. 4th Workshop on Network Coding, Theory, and Applications (NetCod)*. IEEE, 2009, pp. 62–67.

[10] ——, "Random linear network coding for time-division duplexing: field size considerations," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 2009.

[11] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, "Network coding for mobile devices - systematic binary random rateless codes," in *Proc. IEEE Int. Conf. Communications (ICC) workshop*, June 2009.

[12] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010.

[13] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER: Efficient retransmission scheme for wireless LANs," in *Proc. ACM CoNEXT*, Dec. 2007.

[14] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proc. 5th Workshop on Network Coding, Theory, and Applications (NetCod)*, 2009, pp. 80–85.

[15] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. on Wireless Commun. and Netw.*, pp. 1–14, Jan. 2010.

[16] P. Sadeghi, M. Yu, and N. Aboutorab, "On throughput-delay tradeoff of network coding for wireless communications," in *Proc. IEEE Int. Symp. Information Theory and its Applications (ISITA)*, 2014, pp. 689–693.

[17] M. Yu, P. Sadeghi, and N. Aboutorab, "From instantly decodable to random linear network coding," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3943–3955, Oct. 2014.

[18] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[19] M. Luby, "Lt codes," in *Proc. IEEE 54th Annual Symposium on Foundations of Computer Science*, 2002, pp. 271–271.

[20] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[21] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, 2011.

[22] L. Yang, Y. Sagduyu, J. Li, and J. Zhang. Adaptive network coding for scheduling real-time traffic with hard deadlines. [Online]. Available: http://arxiv.org/abs/1203.4008

[23] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Information Theory Workshop, 2007. ITW'07. IEEE*, 2007.

[24] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. ICC*, 2010.

[25] ——, "Coding opportunity densification strategies for instantly decodable network coding," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 5077–5089, 2013.

[26] A. Le, A. S. Tehrani, A. G. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc. IEEE Int. Symp. Network Coding (NetCod)*, 2013.

[27] N. Aboutorab, P. Sadeghi, and S. Sorour, "Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1269–1309, 2014.

[28] S. Sorour and S. Valaee, "Completion delay reduction in lossy feedback scenarios for instantly decodable network coding," in *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011, pp. 2025–2029.

[29] ——, "Completion delay minimization for instantly decodable network coding with limited feedback," in *Proc. IEEE ICC*, 2011.

[30] N. Aboutorab, P. Sadeghi, and S. E. Tajbakhsh, "Instantly decodable network coding for delay reduction in cooperative data exchange systems," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2013, pp. 3095–3099.

[31] M. S. Karim, N. Aboutorab, A. A. Nasir, and P. Sadeghi, "Decoding delay reduction in network coded cooperative systems with intermittent status update," in *Proc. IEEE Information Theory Workshop (ITW)*, 2014, pp. 391–395.

[32] Y. Keshtkarjahromi, H. Seferoglu, R. Ansari, and A. Khokhar, "Content-aware instantly decodable network coding over wireless networks," *arXiv preprint arXiv:1407.1497*, 2014.

[33] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff, *Combinatorics and Graph Theory, 2nd Edition*. Springer Press, 2008.

[34] M. Kubale, *Graph Coloring*, ser. Contemporary Mathematics. American Mathematical Society, 2004.

[35] D. P. Gelle, "Problem 5713," *American Mathematical Monthly*, vol. 77, p. 85, 1970.

[36] W. Xiao and D. Starobinski, "Extreme value FEC for reliable broadcasting in wireless networks," 2009.

[37] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proc. INFOCOM*, 2008.

[38] B. Swapna, A. Eryilmaz, and N. B. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6328–6341, 2013.

[39] B. Ballobás, *Random Graphs*, ser. 2nd edition. Cambridge University Press, 2001.

[40] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 869–878, Jun. 2011.

[41] M. Yu, A. Sprintson, and P. Sadeghi, "On minimizing the average packet decoding delay in network coded wireless broadcast," in *Proc. IEEE Int. Symp. Network Coding (NetCod, submitted to)*, 2015. [Online]. Available: http://arxiv.org/abs/1503.03942

[42] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *Commun. of the ACM*, vol. 16, Sep. 1973.